

WHAT IS CLAIMED IS

1. A method for the secure exchange of objects in a distributed computing
5 environment, comprising:

a user accessing a client device;

10 generating a computer programming language object from a data representation language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device;

15 the user terminating said accessing the client device; and

deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

20 2. The method as recited in claim 1, further comprising the client device receiving a message in the data representation language from a service device in the distributed computing environment prior to said generating a computer programming language object, wherein the message includes the data representation language representation of the object.

25 3. The method as recited in claim 1, wherein said accessing a client device comprises the user coupling an identification device to the client device, wherein the identification device provides identification information of the user to the client device, and wherein said terminating said accessing comprises decoupling the identification device from the client device.

4. The method as recited in claim 3, wherein the identification device is a smart card.

5. The method as recited in claim 1, wherein said accessing a client device
5 comprises the user logging on to the client device by providing user identification to the
client device, and wherein said terminating said accessing comprises the user logging off
the client device.

10 6. The method as recited in claim 1, wherein said generating a computer
programming language object from a data representation language representation of the
object is performed by a virtual machine executing within the client device.

15 7. The method as recited in claim 1, further comprising:

generating a plurality of computer programming language objects from data
representation language representations of the objects; and
20 deleting the plurality of computer programming language objects in response to
said terminating access.

8. The method as recited in claim 1, wherein said data representation language is
eXtensible Markup Language (XML).

25 9. The method as recited in claim 1, wherein said computer programming language
is the Java programming language.

10. A method for the secure exchange of objects in a distributed computing
environment, comprising:

30

a user accessing a client device;

the client device receiving a message in a data representation language from a service device in the distributed computing environment, wherein the 5 message includes a data representation language representation of an object;

determining if the user has access rights to the computer programming language object;

10

if said determining determines the user has access rights to the computer programming language object, generating the object from the data representation language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device; and

15

if said determining determines the user does not have access rights to the computer programming language object, not generating the object.

20

11. The method as recited in claim 10, wherein the message further includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming language object uses the access information.

25

12. The method as recited in claim 10, further comprising:

the user terminating said accessing the client device; and

deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

5 13. The method as recited in claim 12, wherein said accessing a client device comprises the user coupling an identification device to the client device, wherein the identification device provides identification information of the user to the client device, and wherein said terminating said accessing comprises decoupling the identification device from the client device.

10 14. The method as recited in claim 13, wherein the identification device is a smart card.

15 15. The method as recited in claim 12, wherein said accessing a client device comprises the user logging on to the client device by providing user identification to the client device, and wherein said terminating said accessing comprises the user logging off the client device.

16. The method as recited in claim 10, further comprising:

20 the user terminating said accessing the client device; and

storing the computer programming language object in response to said terminating access.

25 17. The method as recited in claim 16, further comprising:

the user accessing the client device subsequent to said storing the object; and

30 accessing the stored object during said accessing the client device.

18. The method as recited in claim 17, further comprising storing access rights information of the user with the object, wherein said accessing the stored object comprises verifying the access rights of the user with the stored access rights information.

5

19. The method as recited in claim 10, wherein said generating a computer programming language object from a data representation language representation of the object is performed by a virtual machine executing within the client device.

10 20. The method as recited in claim 10, wherein said data representation language is eXtensible Markup Language (XML).

21. The method as recited in claim 10, wherein said computer programming language is the Java programming language.

15

22. A device configured to:

accept user input to initiate user access of the device;

20

generate a computer programming language object from a data representation language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the device;

25

terminate said user access; and

delete the computer programming language object in response to said terminating access;

30

wherein the deleted object is not accessible for use by subsequent users of the device.

23. The device as recited in claim 22, further configured to receive a message in the
5 data representation language from a source prior to said generating a computer
programming language object, wherein the message includes the data representation
language representation of the object.

24. The device as recited in claim 22, wherein the device is further configured to:

10 couple to an identification device, wherein the identification device is configured
to provide identification information of the user to the device during said
accepting user input; and

15 decouple from the identification device subsequent to said generating, wherein the
device performs said terminating said user access in response to said
decoupling.

25. The device as recited in claim 24, wherein the identification device is a smart
20 card.

26. The device as recited in claim 22, wherein the device is further configured to
accept user input to initiate said terminating said user access.

25 27. The device as recited in claim 22, wherein the device is further configured to:

generate a plurality of computer programming language objects from data
representation language representations of the objects; and

delete the plurality of computer programming language objects in response to said terminating access.

28. The device as recited in claim 22, comprising:

5

a processor;

a memory;

10 a virtual machine executed by said processor from said memory, wherein said generating is performed by the virtual machine.

29. The device as recited in claim 28,

15 wherein said accepting, said terminating, and said deleting are performed by the virtual machine, wherein the object is stored in the memory subsequent to said generating, and wherein, in said deleting, the object is deleted from the memory.

20 30. The device as recited in claim 22, wherein said data representation language is eXtensible Markup Language (XML).

31. The device as recited in claim 22, wherein said computer programming language is the Java programming language.

25

32. A distributed computing system, comprising:

a client device; and

30

a service device;

wherein the client device is configured to:

5 accept user input to initiate user access of the device;

receive a message in a data representation language from the service device, wherein the message includes a data representation language representation of an object;

10 determine if the user has access rights to the computer programming language object;

15 if said determining determines the user has access rights to the computer programming language object, generate the object from the data representation language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device; and

20 if said determining determines the user does not have access rights to the computer programming language object, not generate the object.

33. The system as recited in claim 32, wherein the message further includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming language object uses the access information.

34. The system as recited in claim 32, wherein the client device is further configured 30 to:

accept user input to terminate said access of the client device; and

5 delete the computer programming language object in response to said terminating
said access, wherein the deleted object is not accessible for use by
subsequent users of the client device.

35. The system as recited in claim 34, wherein the client device is further configured
to:

10 couple to an identification device, wherein the identification device is configured
to provide identification information of the user to the client device during
said accepting user input; and

15 decouple from the identification device subsequent to said generating, wherein the
device performs said terminating said user access in response to said
decoupling.

36. The system as recited in claim 35, wherein the identification device is a smart
20 card.

37. The system as recited in claim 32,

wherein the client device comprises:

25 a memory; and

wherein the client device is further configured to:

30 accept user input to terminate said access of the client device; and

store the computer programming language object to the memory in response to said terminating access.

5 38. The system as recited in claim 37, wherein the client device is further configured to:

accept user input to initiate a second user access of the client device subsequent to said storing the object; and

10 provide access to the stored object during said second user access of the client device.

15 39. The system as recited in claim 38, wherein the client device is further configured to:

store access rights information of the user with the object in the memory; and

20 verify the access rights of the user with the stored access rights information prior to said providing access to the stored object.

40. The device as recited in claim 22, comprising:

a processor;

25 a memory;

a virtual machine executed by said processor from said memory, wherein said generating is performed by the virtual machine, and wherein the object is stored in the memory subsequent to said generating.

41. The system as recited in claim 32, wherein said data representation language is
eXtensible Markup Language (XML).

5 42. The system as recited in claim 32, wherein said computer programming language
is the Java programming language.

10 43. A carrier medium comprising program instructions, wherein the program
instructions are computer-executable to implement:

a user accessing a client device;

15 generating a computer programming language object from a data representation
language representation of the object, wherein the object is an instance of a
class in the computer programming language, and wherein the object is
accessible for use during said accessing the client device;

20 the user terminating said accessing the client device; and

deleting the computer programming language object in response to said
terminating access, wherein the deleted object is not accessible for use by
subsequent users of the client device.

25 44. The carrier medium as recited in claim 43, wherein the program instructions are
further computer-executable to implement the client device receiving a message in the
data representation language from a service device in the distributed computing
environment prior to said generating a computer programming language object, wherein
the message includes the data representation language representation of the object.

30

45. The carrier medium as recited in claim 43, wherein the program instructions are further computer-executable to implement:

5 generating a plurality of computer programming language objects from data representation language representations of the objects; and

deleting the plurality of computer programming language objects in response to said terminating access.

10 46. The carrier medium as recited in claim 43, wherein said data representation language is eXtensible Markup Language (XML), and wherein said computer programming language is the Java programming language.

15 47. A carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a user accessing a client device;

20 the client device receiving a message in a data representation language from a service device in the distributed computing environment, wherein the message includes a data representation language representation of an object;

25 determining if the user has access rights to the computer programming language object;

if said determining determines the user has access rights to the computer programming language object, generating the object from the data representation language representation of the object, wherein the object is

an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device; and

if said determining determines the user does not have access rights to the
5 computer programming language object, not generating the object.

48. The carrier medium as recited in claim 47, wherein the message further includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming language object
10 uses the access information.

49. The carrier medium as recited in claim 47, wherein the program instructions are further computer-executable to implement:

15 the user terminating said accessing the client device; and

deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

20 50. The carrier medium as recited in claim 49, wherein said accessing a client device comprises the user coupling an identification device to the client device, wherein the identification device provides identification information of the user to the client device, and wherein said terminating said accessing comprises decoupling the identification
25 device from the client device.

51. The carrier medium as recited in claim 47, wherein the program instructions are further computer-executable to implement:

30 terminating said accessing the client device; and

storing the computer programming language object in response to said terminating access; and

5 storing access rights information of the user with the object.

52. The carrier medium as recited in claim 51, wherein the program instructions are further computer-executable to implement:

10 the user accessing the client device subsequent to said storing the object;

verifying the access rights of the user with the stored access rights information;
and

15 if said verifying verifies the access rights of the user to the object, allowing access
to the stored object during said accessing the client device.

53. The carrier medium as recited in claim 47, wherein said data representation language is eXtensible Markup Language (XML), and wherein said computer 20 programming language is the Java programming language.